

## INTRODUCCION

---

El funcionamiento básico de una persiana es el siguiente: existe un motor AC sincrónico (normalmente instalado en el propio eje de la persiana) conectado a un pequeño circuito que le indica en qué dirección debe moverse.

Dichos motores poseen 3 cables: 2 de fase (uno para cada sentido de movimiento) y 1 neutro.

Estos motores suelen incorporar un mecanismo llamado "fin de carrera", que consiste en un pequeño interruptor mecánico que detiene el motor cuando la persiana ha bajado en su totalidad o cuando está íntegramente enrollada.

## NOTAS

---

- Los motores utilizados comúnmente en la automatización de persianas son de tipo "AC síncrono". Debemos hacer un sistema compatible con este tipo de motores.

\* Estos motores son de velocidad constante determinada por la frecuencia de la corriente alterna que lo alimenta.

- He encontrado algunos sitios donde dicen montar motores DC (pero no parece algo muy común).

## POSIBLES DISEÑOS

---

### 1. Servomotor

Para conseguir el giro ilimitado del servo hay que modificarlo

- Eliminar los topes mecánicos de las ruedas dentadas
- Sustituir el potenciómetro/encoder por uno multivuelta

\* DESCARTADO. Puesto que esto es una solución poco flexible.

### 2. DC-Relé-Sensor

El motor DC alimentado por tensión constante podría ser sustituido sin ningún problema por un motor AC sincronizado. Los motores AC síncronos tienen una fase para controlar cada sentido de giro y giran a velocidad constante. En nuestro diseño con motor DC alimentado con tensión constante (velocidad de giro constante) simularemos (de alguna manera) estas 2 fases de control de cada sentido de giro de manera que la interfaz de cara al controlador del motor sea la misma.

Material necesario:

1 x Motor DC con alimentación externa (pila o transformador) constante. Por ej, 3V o 5V

2 x Relé o 1 x Relé doble - Para invertir la polaridad del motor

¿2 x Potenciómetro - Para controlar los topes de giro del motor?

\* ESTE ES EL DISEÑO QUE IMPLEMENTARÉ

## PROBLEMAS

---

1. No disponemos de un motor real de persiana, por lo que haremos un prototipo que sea lo más parecido posible.

En un principio únicamente disponíamos de pequeños motores DC, pero posteriormente conseguimos hacernos con un motor AC síncrono (sin finales de carrera). El problema es que ninguno de estos motores poseen los mecanismos de finales de carrera.

Los motores tubulares que suelen utilizarse en automatización de persianas ya tienen un mecanismo integrado que para el motor automáticamente al llegar a unos puntos regulables. Por

tanto, vamos a suponer que los motores tienen integrado dicho mecanismo.

2. ¿Como podemos detectar el estado de la persiana desde la aplicación?. Como el usuario interactúa de forma externa al sistema, no podemos saber su estado.

Tenemos varias opciones

+ Utilizar contadores sincronizados con los tiempos de subida y bajada de la persiana. POCO PRECISO Y POCO SEGURO.

+ Instalación de un potenciómetro que nos indique la posición exacta de la persiana. Habría que utilizar un sistema de poleas para ajustar las vueltas del cajetín a las vueltas del potenciómetro. DEMASIADO APARATOSO.

+ Instalación de un encoder como el de la rueda de un ratón (los hay mecánicos y ópticos). Pero requiere tiempo destripar el ratón.

\* Este diseño nos daría una precisión muy grande de posicionamiento de la persiana. Pero no podríamos controlar el estado real de la persiana. Mediante los reed switch podemos detectar si la persiana ha llegado realmente a una posición determinada o no. ES MAS SEGURO ASI.

+ Instalación de sensor magnético que vaya detectando imanes empotrados en las lamas de la persiana.

+ Sensor de efecto Hall o Reed Switch:

- Sensor en cajetín de la persiana e Imán en eje del tambor de la persiana. Nos indica el numero de vueltas que ha dado.

Esta opción tiene el problema de que cada vuelta de eje no corresponde a una longitud de persiana constante.

- Imanes en lamas de la persiana & Sensor en riel de la persiana. Nos indica el avance exacto de la persiana (creo que podemos detectar el sentido de avance también).

Podemos incluir tantos imanes como queramos en las lamas para tener un control ajustado a las necesidades del usuario.

+ DECISION: Me he decantado por reed switches ya que parece que es menos complicado elegir el adecuado (sensores hall hay de tres tipos y en las tiendas online no suelen indicar de que tipo es el que compras)

+ IMPORTANTE: No olvidar distinguir el estado "Persiana compactada" = Todas las lamas apoyadas unas sobre otras.

3. ¿Como mantenemos el control manual?

Depende de la interfaz de control de que disponga la persiana:

- Interruptores mecánicos



+ Estos mecanismos son como llaves de luz dobles con un mecanismo físico adicional que impide que las dos teclas cierren el circuito simultáneamente.

+ Nos darán problemas, ya que la activación de una tecla es permanente de manera que dicha fase del motor se queda activa también de forma permanente. Y si colocamos los relés

controlados por Arduino en paralelo no funcionaría porque las llaves mecánicas podrían estar fijas en una posición que activase una fase impidiendo que podamos activar la contraria sin estropear el motor.

+ La única opción sería hacer dobles circuitos para controlar cada fase para poder introducir relés adicionales en cada fase que nos permitan activar o desactivar las fases sin dañar el sistema. El problema es que tendríamos un comportamiento un tanto desconcertante de la llave física, ya que no sabríamos si la fase estaría activa en su posición superior o en su posición inferior (al igual que ocurre con llaves conmutadas que controlan la iluminación en un pasillo).

+ Tras todo lo anterior pienso que las mejores opciones serían:

1. Utilizar un switch SPDT con centro OFF → (ON) OFF (ON). Este es el tipo de interruptor es mas seguro, ya que impide completamente accionar las dos llaves a la vez.

2. Sustituirla por una digital en la que todas las acciones se gestionen por un chisme programado a través de una interfaz de pulsadores (2 o 3).

\* Esta podría estar gestionada por un controlador independiente o por el propio Arduino que controle la persiana.

- Interruptor digital



+ Es mas facil de acoplar al nuestro sistema.

4. Necesitamos simular la interfaz de un motor común de persiana (AC con 3 entradas: Fase subida, Fase bajada, Común) para poder hacer un sistema flexible

5. Control de obstáculos

Podemos servirnos de sensores como:

+ fotocélula con un reflector como la puerta de un garaje o de un ascensor

+ láser apuntando a una resistencia LDR o un fototransistor ajustada para recibir umbral que no interfiera con luz solar

## FUNCION AVANZADA

Programar apertura y cierra de persiana en función de sensor de luz ventana y/o temperatura de la sala con objeto de regular la luz y la temperatura de la sala.